Statystyka w R

Magda G.

13 sierpnia 2018

## Statystyka opisowa

##### Funkcje statystyki opisowej w R

Opis

Funkcja\_R

Średnia

mean()

Odchylenie standardowe

sd()

Wariancja

var()

Minimum

min()

Maksimum

maximum()

Mediana

median()

Moda

mfv()

Zakres wartości

range()

Kwantyle próby

quantile()

Rozstęp ćwiartkowy

IQR()

Funkcja generyczna

summary()

W celu przedstawienia możliwości R użyjemy danych dostarczanych przez R - iris

dane <- iris # Przypisanie danych do zmiennej  
head(dane, 6)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
1 5.1 3.5 1.4 0.2 setosa  
2 4.9 3.0 1.4 0.2 setosa  
3 4.7 3.2 1.3 0.2 setosa  
4 4.6 3.1 1.5 0.2 setosa  
5 5.0 3.6 1.4 0.2 setosa  
6 5.4 3.9 1.7 0.4 setosa

#### Miary skupienia

Dla pojedynczej próby do określania tendencji centralnej danych służą średnia (wrażliwa na wartości odstające, mediana (bardziej ‘odporna’ alternatywa dla średniej) i moda (najczęstsza wartość w próbie)

mean(dane$Sepal.Length) # średnia

[1] 5.843333

median(dane$Sepal.Length) # mediana

[1] 5.8

# install.packages("modeest")  
require(modeest)  
mfv(dane$Sepal.Length) # moda

[1] 5

#### Miary zmienności

Opisują rozproszenie wartości cechy w populacji/ próbie. Są miarą dyspersji danych.

1. minimum, maksimium i zakres danych.

###### Używane rzadko ponieważ są bardzo wrażliwe na dane odstające.

min(dane$Sepal.Length) # minimum

[1] 4.3

max(dane$Sepal.Length) # maksimum

[1] 7.9

range(dane$Sepal.Length) # zakres danych

[1] 4.3 7.9

1. Kwantyle

###### Kwartyle dzielą dane na 4 zakresy. Rozstęp ćwiartkowy (różnica pomiędzy 1 i 3 kwartylem w pewnych sytuacjach zastępuje odchylenie standardowe. Kwartyle są szczególnymi przypadkami kwantyli.

quantile(x, probs = seq(0, 1, 0.25))

x: wektor liczbowy, w którym określamy kwantyle probs: liczbowy wektor prawdopodobieństw o zakresie [0,1].

###### Rozstęp ćwiartkowy używany jest dość często, szczególnie w połączeniu z medianą. Są mało wrażliwe na wartości odstające.

quantile(dane$Sepal.Length) # standardowo funkja zwraca kwartle

0% 25% 50% 75% 100%   
 4.3 5.1 5.8 6.4 7.9

quantile(dane$Sepal.Length, seq(0, 1, 0.1)) # funkcja zwróci decyle

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%   
4.30 4.80 5.00 5.27 5.60 5.80 6.10 6.30 6.52 6.90 7.90

IQR(dane$Sepal.Length) # zwraca zakres ćwiartkowy

[1] 1.3

1. Wariancja i odchylenie standardowe

###### Wariancja to średni kwadrat odchyleń od średniej. Używana jest bardzo rzadko, głównie w kalkulacjach, ponieważ nie ma tych samych jednostek, co próba. Odchylenie standardowe jest pierwiastkiem z wariancji. Jest miarą przecietnego odchylenia wartości w próbie od średniej. Używane jest, gdy miarą tendencji centralnej jest średnia.

var(dane$Sepal.Length) # wariancja

[1] 0.6856935

sd(dane$Sepal.Length) # odchylenie standardowe

[1] 0.8280661

1. Odchylenie medianowe (MAD)

###### Mierzy odchylenie wartości od wyznaczonej mediany. Używana dość rzadko, ale jest mało wrażliwa na wartości odstające.

median(dane$Sepal.Length)

[1] 5.8

mad(dane$Sepal.Length)

[1] 1.03782

#### Funkcja **summary()**

###### Służy do policzenia statystyki opisowej 1 zmiennej, albo całej ramki danych

summary(dane$Sepal.Length) # podsumowanie statystyki opisowej zmiennej

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 4.300 5.100 5.800 5.843 6.400 7.900

summary(dane, digits = 2) # podsumowanie statystyki opisowej ramki danych

Sepal.Length Sepal.Width Petal.Length Petal.Width Species   
 Min. :4.3 Min. :2.0 Min. :1.0 Min. :0.1 setosa :50   
 1st Qu.:5.1 1st Qu.:2.8 1st Qu.:1.6 1st Qu.:0.3 versicolor:50   
 Median :5.8 Median :3.0 Median :4.3 Median :1.3 virginica :50   
 Mean :5.8 Mean :3.1 Mean :3.8 Mean :1.2   
 3rd Qu.:6.4 3rd Qu.:3.3 3rd Qu.:5.1 3rd Qu.:1.8   
 Max. :7.9 Max. :4.4 Max. :6.9 Max. :2.5

#### Funkcja **sapply()**

###### Korzystając z funkcji sapply() możemy zastosować określoną funkcję do listy wektorów. Przykładowo dla każdej kolumny ramki danych możemy policzyć średnią, odchylenie stnadardowe, wariancję, minimum, kwantyle i tp.

sapply(dane[, -5], mean) # liczy średnią z kolumn

Sepal.Length Sepal.Width Petal.Length Petal.Width   
 5.843333 3.057333 3.758000 1.199333

sapply(dane[, -5], quantile) # liczy kwartyle dla kolumn

Sepal.Length Sepal.Width Petal.Length Petal.Width  
0% 4.3 2.0 1.00 0.1  
25% 5.1 2.8 1.60 0.3  
50% 5.8 3.0 4.35 1.3  
75% 6.4 3.3 5.10 1.8  
100% 7.9 4.4 6.90 2.5

#### Funkcja **stat.desc()**

###### Funkcja stat.desc() dostarcza kilka innych statystych niż wcześniej wspomniane. Liczy również błąd standardowy średniej, przedział ufności dla średniej (standardowo dla p = 0,95) i współczynnik wariancji (odchylenie standardowe podzielone przez średnią)

# install.packages("pastecs")  
library(pastecs)  
res <- stat.desc(dane[, -5])  
round(res, 2)

Sepal.Length Sepal.Width Petal.Length Petal.Width  
nbr.val 150.00 150.00 150.00 150.00  
nbr.null 0.00 0.00 0.00 0.00  
nbr.na 0.00 0.00 0.00 0.00  
min 4.30 2.00 1.00 0.10  
max 7.90 4.40 6.90 2.50  
range 3.60 2.40 5.90 2.40  
sum 876.50 458.60 563.70 179.90  
median 5.80 3.00 4.35 1.30  
mean 5.84 3.06 3.76 1.20  
SE.mean 0.07 0.04 0.14 0.06  
CI.mean.0.95 0.13 0.07 0.28 0.12  
var 0.69 0.19 3.12 0.58  
std.dev 0.83 0.44 1.77 0.76  
coef.var 0.14 0.14 0.47 0.64

Brakujące wartości NA w danych moga zaburzyć działanie niektórych funkcji, dlatego często stosuje się parametr na.rm = TRUE, który informuje funkcje, że ma usunąć wartości NA, przed wykonaniem obliczeń

mean(dane$Sepal.Length, na.rm = TRUE)

[1] 5.843333

#### Statystyka opisowa danych podzielonych na kategorie

Liczenie parametrów dla grupdanych możliwe jest dzięki funkcjom group\_by() i summarise() znajdujących sie w pakiecie dplyr

# install.packages("dplyr")  
library(dplyr)  
group\_by(dane, Species) %>% # Podział danych na gatunki  
summarise(  
 count = n(), # zliczenie danych w kazdej z grup  
 mean = mean(Sepal.Length, na.rm = TRUE), # średnia w grupach  
 sd = sd(Sepal.Length, na.rm = TRUE) # odchylenie standardowe w grupach  
 ) # wartości dotyczą długości płatków

# A tibble: 3 x 4  
 Species count mean sd  
 <fct> <int> <dbl> <dbl>  
1 setosa 50 5.01 0.352  
2 versicolor 50 5.94 0.516  
3 virginica 50 6.59 0.636

Operator %>% służy do tworzenia ciągów analiz. Jest to mozliwe dzięki pakietowi dplyr, ponieważ R wymaga zapisywania danych cząstkowych i wykorzystywania ich na kolejnych etapach analizy. Zastosowanie %>% usprawnia analizę danych w R oraz upodabnia funkcjonalnie R do innych języków peogramowania np.: bash czy Python.

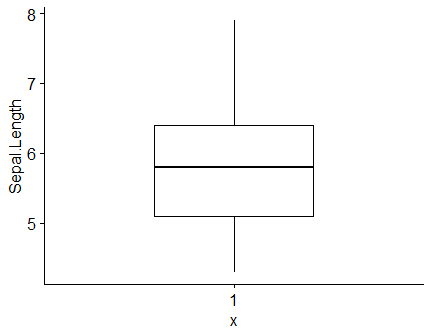
### Graficzne przedstawienie parametrów próby

Będziemy pracować przede wszystkim korzystając z funkcji pakietu ggpubr oraz danych iris wprowadzonych wcześniej jako "dane"

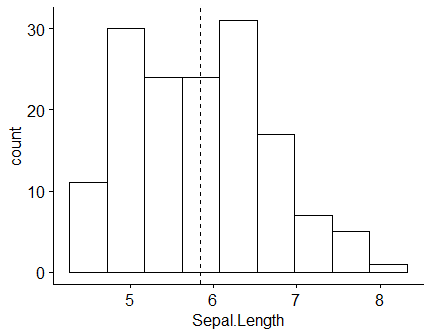
# Instalacja ggpubr - jeśli wymagana `od'hash'uj` linijki  
# if(!require(devtools)) install.packages("devtools")  
# devtools::install\_github("kassambara/ggpubr")  
# lub instalacja z CRAN  
# install.packages("ggpubr")  
  
library(ggpubr)

Kilka typów wykresów o różnych zastosowaniach

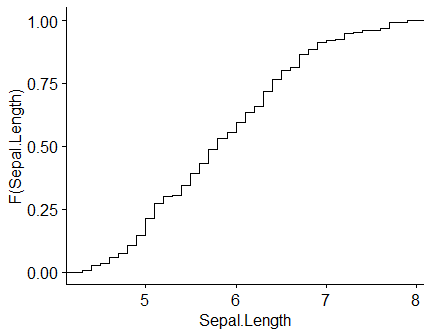
# Wykres pudełkowy - pokazuje dystrybucję danych  
ggboxplot(dane, y = "Sepal.Length", width = 0.5)



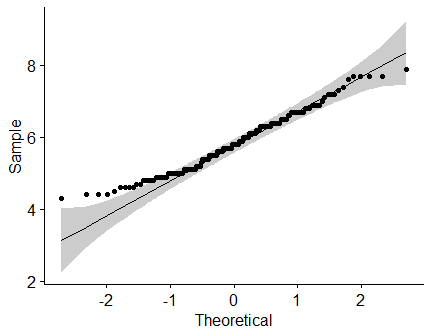
# Histogram - pokazuje liczbe obserwacji w danym zakresie wartości (bin)   
# dodatkowo na wykresie zaznzaczono średnią  
gghistogram(dane, x = "Sepal.Length", bins = 9, add = "mean")



#ECDF (Empirical cumulative distribution function) - frakcja danych o wartości mniejszej niż x lub równej  
ggecdf(dane, x = "Sepal.Length")

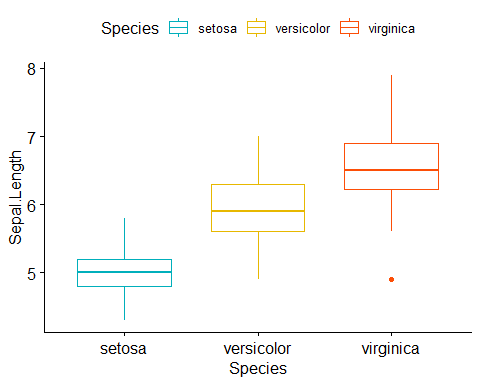


# Q-Q plots - służy do sprawdzenia czy dane mają rozkład normalny  
ggqqplot(dane, x = "Sepal.Length")

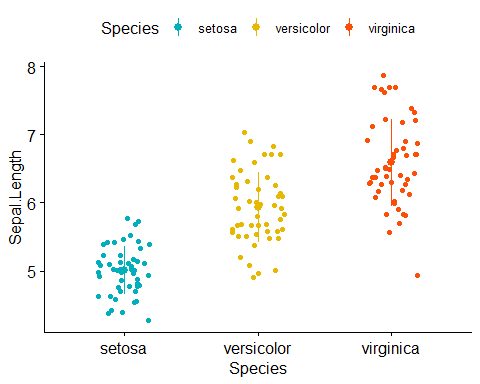


#### Graficzne przedstawienie danych pogrupowanych na kategorie

library("ggpubr")  
# Wykres pudełkowy z podziałem na gatunki oraz zależnym od gatunku kolorem  
ggboxplot(dane, x = "Species", y = "Sepal.Length",  
 color = "Species",  
 palette = c("#00AFBB", "#E7B800", "#FC4E07"))



# Stripchart z podziałem na gatunki oraz zależnym od gatunku kolorem  
ggstripchart(dane, x = "Species", y = "Sepal.Length",  
 color = "Species",  
 palette = c("#00AFBB", "#E7B800", "#FC4E07"),  
 add = "mean\_sd")



Kiedy dancyh w grupach jest mało lepiej jest używać stripchart niż wykresów pudełkowych.

### Tabele częstości/ kontygencji

Służą do przedstawiania liczebności danych kategorycznych. W R do ich utworzenia służy polecenie table()

Do analizy posłużą dane “HairEyeColor” zawierające dane o płci, kolorze włosów i oczu 592 studentów

# Utworzenie ramki danych  
df <- as.data.frame(HairEyeColor)  
kolor\_oczu\_wlosow <- df[rep(row.names(df), df$Freq), 1:3]   
# Przeksztalca dane tworząc liczbę wierszy każdej z kombinacji kategorii  
# równą podanej w 4 kolumnie (df$Freq) częstotliwości  
rownames(kolor\_oczu\_wlosow) <- 1:nrow(kolor\_oczu\_wlosow)  
head(kolor\_oczu\_wlosow)

Hair Eye Sex  
1 Black Brown Male  
2 Black Brown Male  
3 Black Brown Male  
4 Black Brown Male  
5 Black Brown Male  
6 Black Brown Male

# Utworzenie czynników (factor) kategorii  
Hair <- kolor\_oczu\_wlosow$Hair  
Eye <- kolor\_oczu\_wlosow$Eye  
levels(Eye) # Poziomy czynnika 'Eye' kolor oczu

[1] "Brown" "Blue" "Hazel" "Green"

levels(Hair) # Poziomy czynnika 'Hair' kolor włosów

[1] "Black" "Brown" "Red" "Blond"

# Prosta dystrybucja częstotliwości dla 1 zmiennej - koloru włosów  
table(Hair)

Hair  
Black Brown Red Blond   
 108 286 71 127

# Prosta dystrybucja częstotliwości dla 1 zmiennej - koloru oczu  
table(Eye)

Eye  
Brown Blue Hazel Green   
 220 215 93 64

# 2-kierunkowa tabela dystrybucji częstotliwości dla kombinacji obu zmiennych  
table(Hair, Eye)

Eye  
Hair Brown Blue Hazel Green  
 Black 68 20 15 5  
 Brown 119 84 54 29  
 Red 26 17 14 14  
 Blond 7 94 10 16

xtabs(~ Hair + Eye, data = kolor\_oczu\_wlosow) # Tworzy tabelę krzyżową na podstawie podanej formuły

Eye  
Hair Brown Blue Hazel Green  
 Black 68 20 15 5  
 Brown 119 84 54 29  
 Red 26 17 14 14  
 Blond 7 94 10 16

# Wprowadzamy dodatkowo podział na kobiety i mężczyzn (3 zmienna)  
xtabs(~Hair + Eye + Sex, data = kolor\_oczu\_wlosow)

, , Sex = Male  
  
 Eye  
Hair Brown Blue Hazel Green  
 Black 32 11 10 3  
 Brown 53 50 25 15  
 Red 10 10 7 7  
 Blond 3 30 5 8  
  
, , Sex = Female  
  
 Eye  
Hair Brown Blue Hazel Green  
 Black 36 9 5 2  
 Brown 66 34 29 14  
 Red 16 7 7 7  
 Blond 4 64 5 8

# Funkcja ftable() tworzy spłaszczoną (ładniejszą) wersję tabeli krzyżowej  
ftable(Sex + Hair ~ Eye, data = kolor\_oczu\_wlosow)

Sex Male Female   
 Hair Black Brown Red Blond Black Brown Red Blond  
Eye   
Brown 32 53 10 3 36 66 16 4  
Blue 11 50 10 30 9 34 7 64  
Hazel 10 25 7 5 5 29 7 5  
Green 3 15 7 8 2 14 7 8

Możliwe jest również policznie sum w kolumnach i rzędach oraz relatywnej częstości (stosunek wartości w danej pozycji tabeli do zliczonej wartości w kolumnie lub rzędzie). Funkcja margin.table() zlicza wartości w kulumach/ rzędach, a prop.table() wylicza częstości relatywne.

###### **Format funkcji:**

margin.table(x, margin = NULL)

prop.table(x, margin = NULL)

x: utworzona tabela margin: indeks (1 dla rzędów and 2 dla kolumn)

# Oblicznie wartości brzegowych  
oba.tbl <- table(Hair, Eye) # Tabela krzyżowa dla obu wartości  
  
# Wartości brzegowe rzdów  
margin.table(oba.tbl, 1) # Wartości brzegowe rzdów

Hair  
Black Brown Red Blond   
 108 286 71 127

margin.table(oba.tbl, 2) # Wartości brzegowe kolumn

Eye  
Brown Blue Hazel Green   
 220 215 93 64

prop.table(oba.tbl, 1) # Częstości w odniesieniu do wartości brzegowych rzędów

Eye  
Hair Brown Blue Hazel Green  
 Black 0.62962963 0.18518519 0.13888889 0.04629630  
 Brown 0.41608392 0.29370629 0.18881119 0.10139860  
 Red 0.36619718 0.23943662 0.19718310 0.19718310  
 Blond 0.05511811 0.74015748 0.07874016 0.12598425

round(prop.table(oba.tbl, 1), 3)\*100 # Podanie wartości w procentach

Eye  
Hair Brown Blue Hazel Green  
 Black 63.0 18.5 13.9 4.6  
 Brown 41.6 29.4 18.9 10.1  
 Red 36.6 23.9 19.7 19.7  
 Blond 5.5 74.0 7.9 12.6

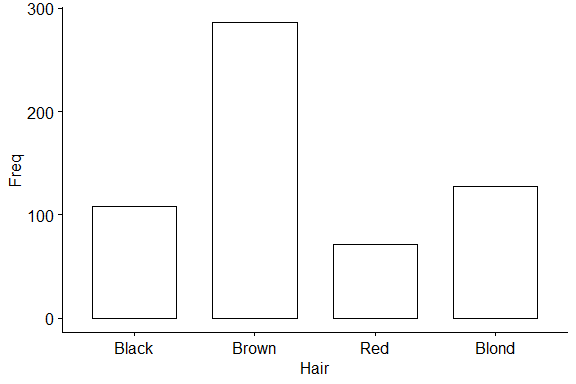
# funkcja round() zaokrągla wartości do podanej liczby miejsc po przcinku

#### Graficzne przedstawienie wyników

# Przekonwertowanie tabeli na ramkę danych  
dfh <- as.data.frame(table(Hair))  
dfh

Hair Freq  
1 Black 108  
2 Brown 286  
3 Red 71  
4 Blond 127

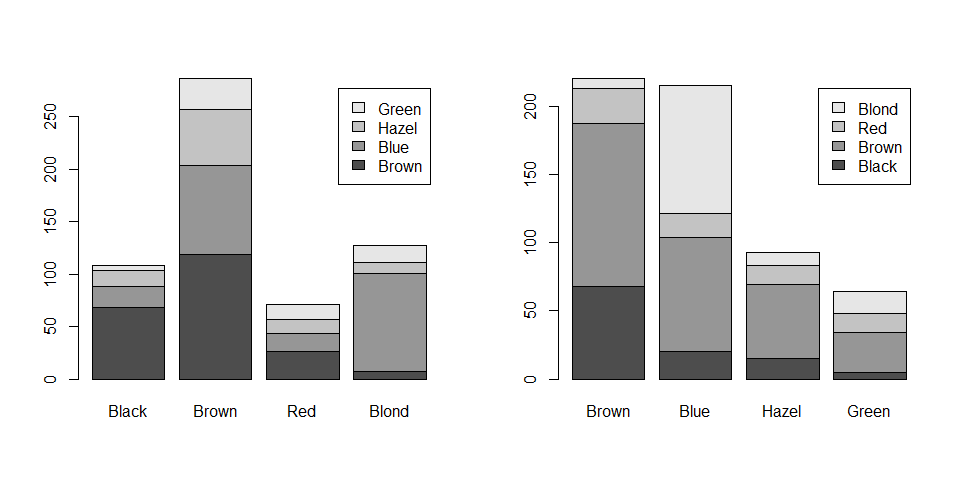
# Utworzenie wykresu słupkowego  
library(ggpubr) # ładowanie pakietu ggpubr  
# wykorzystanie możliwości pakietu do tworzenia wykresów bazującego na uproszczonym działaniu pakietu ggplot2  
ggbarplot(dfh, x = "Hair", y = "Freq")



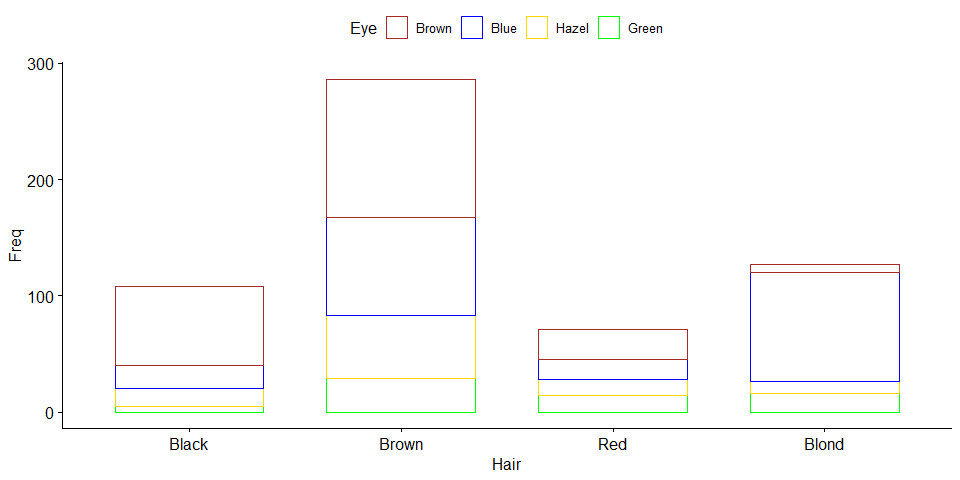
dfb <- as.data.frame(xtabs(~ Hair + Eye, data = kolor\_oczu\_wlosow))  
head(dfb)

Hair Eye Freq  
1 Black Brown 68  
2 Brown Brown 119  
3 Red Brown 26  
4 Blond Brown 7  
5 Black Blue 20  
6 Brown Blue 84

# Twrzenie wykresu słupkowego 1 zmiennej z uwzględnieniem 2 zmiennej  
par(mfrow=c(1,2))  
barplot(xtabs(~ Eye + Hair, data = kolor\_oczu\_wlosow), legend = levels(Eye))  
barplot(xtabs(~ Hair + Eye, data = kolor\_oczu\_wlosow), legend = levels(Hair))



# library(ggpubr) - usunąć #, jesli w danej sesji nie był używany ten pakiet  
ggbarplot(dfb, x = "Hair", y = "Freq",  
 color = "Eye",   
 palette = c("brown", "blue", "gold", "green"))



# Wersja z osobnymi słupkami dla częstotliwości  
ggbarplot(df, x = "Hair", y = "Freq",  
 color = "Eye", position = position\_dodge(),  
 palette = c("brown", "blue", "gold", "green"))

